12-2014

High-dimensional Data Stream Classification via Sparse Online Learning

Dayong WANG
Nanyang Technological University, Singapore

Pengcheng WU
Singapore Management University, pcwu@smu.edu.sg

Peilin ZHAO
ASTAR

Yue WU
University of Science and Technology of China

Chunyan MIAO
Nanyang Technological University

See next page for additional authors

DOI: https://doi.org/10.1109/ICDM.2014.46

High-dimensional Data Stream Classification via Sparse Online Learning

Dayong Wang*, Pengcheng Wu†, Peilin Zhao‡, Yue Wu§, Chunyan Miao* and Steven C.H. Hoi†
*School of Computer Engineering, Nanyang Technological University, 639798, Singapore
†School of Information Systems, Singapore Management University, 178902, Singapore
‡Institute for Infocomm Research, A*STAR, 138632, Singapore
§University of Science and Technology of China, Hefei, 230026, China
Email: {dywang, ascymiao}@ntu.edu.sg, {pcwu, chhoi}@smu.edu.sg, zhaop@i2r.a-star.edu.sg, wye@mail.ustc.edu.cn

***Abstract*—The amount of data in our society has been exploding in the era of big data today. In this paper, we address several open challenges of big data stream classification, including high volume, high velocity, high dimensionality, and high sparsity. Many existing studies in data mining literature solve data stream classification tasks in a batch learning setting, which suffers from poor efficiency and scalability when dealing with big data. To overcome the limitations, this paper investigates an online learning framework for big data stream classification tasks. Unlike some existing online data stream classification techniques that are often based on first-order online learning, we propose a framework of Sparse Online Classification (SOC) for data stream classification, which includes some state-of-the-art first-order sparse online learning algorithms as special cases and allows us to derive a new effective second-order online learning algorithm for data stream classification. We conduct an extensive set of experiments, in which encouraging results validate the efficacy of the proposed algorithms in comparison to a family of state-of-the-art techniques on a variety of data stream classification tasks.**

***Keywords*-data stream classification; sparse; online learning;**

I. Introduction

In the era of big data today, the amount of data in our society has been exploding, which has raised many opportunities and challenges for data analytic research in data mining community. In this work, we aim to address the challenging real-world big data stream classification task, such as web-scale spam email classification. In general, big data stream classification has several characteristics:

- **high volume**: one has to deal with huge amount of existing training data, in million or even billion scale;
- **high velocity**: new data often arrives sequentially and very rapidly, e.g., about 182.9 billion emails are sent/received worldwide every day according to an email statistic report by the Radicati Group [1];
- **high dimensionality**: there are a large number of features, e.g., for some spam email classification tasks, the length of the vocabulary list can go up from $10,000$ to $50,000$ or even to million scale;
- **high sparsity**: many feature elements are zero, and the faction of active features is often small, e.g., the spam email classification study in [2] showed that accuracy saturates with dozens of features out of tens of thousands of features.

The above characteristics present huge challenges for big data stream classification tasks when using conventional data stream classification techniques that are often restricted to batch learning setting. To tackle the above challenges, a promising approach is to explore online learning methodology that performs incremental training over streaming data in a sequential manner. In contrast to batch learning algorithms, online algorithms are not only more efficient and scalable, but also able to avoid expensive re-training cost when handling new training data. However, the traditional online-learning algorithms suffer from critical limitation for high-dimensional data. This is because they assume at least one weight for every feature and most of the learned weights are often nonzero, making them of low efficiency not only in computational time but also in memory cost for both training and test phases. Sparse online learning [3] aims to overcome this limitation by inducing sparsity in the weights learned by an online-learning algorithm.

In this paper, we introduce a framework of Sparse Online Learning for solving large-scale high-dimensional data stream classification tasks. We show that the proposed framework covers some existing first-order sparse online classification algorithm, and is able to further derive new algorithms by exploiting the second order information. The proposed sparse online classification scheme is far more efficient and scalable than the traditional batch learning algorithms for data stream classification tasks. We further give theoretical analysis of the proposed algorithm and conduct an extensive set of experiments. The empirical evaluation shows that the proposed algorithm could achieve state-of-the-art performance. The rest of this paper is organized as follows. Section 2 reviews related work. Section 3 presents our problem formulation. Section 4 proposes our novel framework . Section 5 discusses our experimental results, and section 6 concludes this work.

II. Related Work

A. Sparse Online Learning

Online learning represents a family of efficient and scalable machine learning algorithms [4]. The general online

learning algorithms have solid theoretical guarantees and perform well on many applications. However, they exploit the full features, which is not suitable for large-scale high-dimensional problem. To tackle this limitation, the *sparse online learning* [5, 3, 6] has been extensively studied recently. *Sparse online learning* aims to learn a sparse linear classifier, which only contains limited size of active features. It has been actively studied [5, 7, 8, 9]. There are two group of solutions for *sparse online learning*. The first group study on sparse online learning follows the general idea of subgradient descent with truncation. For example, Duchi and Singer propose the FOBOS algorithm [5], which extends the *Forward-Backward Splitting* method to solve the sparse online learning problem in two phases: (i) an unconstrained subgradient descent step with respect to the loss function, and (ii) an instantaneous optimization for a trade-off between minimizing ℓ_1 norm regularization and keeping close to the result obtained in the first phase. The optimization problem in the second phase can be efficiently solved by adopting simple *soft-thresholding* operations that perform some truncation on the weight vectors. Following the similar scheme, Langford et al. [3] argue that truncation on every iteration is too aggressive as each step modifies the coefficients by only a small amount, and propose the *Truncated Gradient* (TG) method which truncates coefficients every K steps when they are less than a predefined threshold θ. The second group study on sparse online learning mainly follows the dual averaging method of [10], can explicitly exploit the regularization structure in an online setting. For example, One representative work is *Regularized Dual Averaging*(RDA) [7], which learns the variables by solving a simple optimization problem that involves the running average of all past subgradients of the lost functions, not just the subgradient in each iteration. Lee et al. [11] further extends the RDA algorithm by using a more aggressive truncation threshold and generates significantly more sparse solutions.

B. Second-order Online Learning

The general online learning algorithms only exploit the first order information and all features are adopted the same learning rate. This problem can be addressed by *second order online learning* algorithms [12], which aims to dynamically incorporate knowledge of observed data in earlier iteration to perform more informative gradient-based learning. Unlike first order algorithms that often adopt the same learning rate for all coordinates, the second order online learning algorithms adopt different distills to the step size employed for each coordinate. A variety of second order online learning algorithms have been proposed recently. Some technique attempts to incorporates knowledge of the geometry of the data observed in earlier iterations to perform more effective online updates. For example, Balakrishnan et al. [13] propose algorithms for sparse linear classifiers in the massive data setting, which requires $O(d^2)$ time and $O(d^2)$ space in the worst case. Another state-of-the-art technique for second order online learning is the family of confidence-weighted (CW) learning algorithms [14, 15, 16, 17, 18], which exploit confidence of weights when making updates in online learning processes. In general, the second order algorithms are more accurate, converge faster, but fall short in two aspects (i) they incur higher computational cost especially when dealing with high-dimensional data; and (ii) the weight vectors learned are often not sparse, making them unsuitable for high-dimensional data. Recently, Duchi et al. address the sparsity and second order update in the same framework, and proposed the Adaptive Subgradient method [19] (Ada-RDA), which adaptively modifies the proximal function at each iteration to incorporate knowledge about geometry of the data.

III. Sparse Online Learning for Data Stream Classification

A. General Sparse Online Learning

Without loss of generality, we consider the sparse online learning algorithm for the binary classification problem, which is also mentioned as sparse online classification problem in this paper. The sparse online classification algorithm generally works in rounds. Specifically, at the round t, the algorithm is presented one instance $\mathbf{x}_t \in \mathbb{R}^d$, then the algorithm predicts its label as

$$\hat{y}_t = \text{sign}(\mathbf{w}_t^\top \mathbf{x}_t),$$

where $\mathbf{w}_t \in \mathbb{R}^d$ is linear classifier maintained by the algorithm. After the prediction, the algorithm will receive the true label $y_t \in \{+1, -1\}$, and suffer a loss $\ell_t(\mathbf{w}_t)$. Then, the algorithm would update its prediction function $\mathbf{w}_t$ based on the newly received $(\mathbf{x}_t, y_t)$. The standard goal of online learning is to minimize the number of mistakes suffered by the online algorithm. To facilitate the analysis, we firstly introduce several functions. Firstly, the hinge loss $\ell_t(\mathbf{w}; (\mathbf{x}_t, y_t)) = [1 - y_t \mathbf{w}^\top \mathbf{x}_t]_+$, where $[a]_+ = \max(a, 0)$, is the most popular loss function for binary classification problem. Let $\Phi_t, t = 1, \ldots, T$ be δ-strongly convex functions with respect to the norms $\|\cdot\|_{\Phi_t}$ and let $\|\cdot\|_{\Phi_t}^*$ be the respective dual norms. The proposed general sparse online classification (SOC) algorithm is shown in Algorithm 1.

For the proposed general sparse online learning (SOL) algorithm, if ℓ is convex and $\eta_t = \eta$, we can achieve that the regret $R_T = \sum_{t=1}^T \ell_t(\mathbf{w}_t) - \min_{\mathbf{w}} \sum_{t=1}^T \ell_t(\mathbf{w})$ of the proposed framework (1) satisfies the following inequality

$$R_T \leq \frac{\Phi_T(\mathbf{w})}{\eta} + \sum_{t=1}^T [\frac{\eta}{2\delta} \|\mathbf{z}_t\|_{\Phi_t^*}^2 + \lambda_t \|\mathbf{z}_t\|_1] + \frac{\sum_{t=1}^T \Delta_t^*}{\eta} \quad (1)$$

where $\Delta_t^* = \Phi_t^*(\theta_t) - \Phi_{t-1}^*(\theta_t)$. Due to space limitations, we skip the derivation. Given this framework and these analysis, we would drive some specific algorithms and their regret bounds.

Algorithm 1 General Sparse Online Learning (SOL)

INPUT :λ, η
INITIALIZATION : $\theta_1 = 0$.
for $t = 1, \ldots, T$ **do**
  receive $\mathbf{x}_t \in \mathbb{R}^n$
  $\mathbf{u}_t = \nabla \Phi_t^*(\theta_t)$
  $\mathbf{w}_t = \arg\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{u}_t - \mathbf{w}\|_2^2 + \lambda_t \|\mathbf{w}\|_1$
  predict $\hat{y}_t = \text{sign}(\mathbf{w}_t^\top \mathbf{x}_t)$
  receive $y_t \in \{-1, 1\}$ and suffer $\ell_t(\mathbf{w}_t) = [1 - y_t \mathbf{w}_t^\top \mathbf{x}_t]_+$
  $\theta_{t+1} = \theta_t - \eta_t \mathbf{z}_t$ where $\mathbf{z}_t = \nabla \ell_t(\mathbf{w}_t)$;
end for

IV. Derived Algorithms

In this section, we will first recover the RDA [7] algorithm and then derive algorithm utilizing the second order-information. In this section, we will adopt the hinge loss function and denote $\mathcal{L} = \{t | \ell_t(\mathbf{w}_t) > 0\}$ and $L_t = \mathbb{I}_{(\ell_t(\mathbf{w}_t) > 0)}$, where $\mathbb{I}_v$ is indicator function, $\mathbb{I}_v = 1$ if v is true, otherwise $\mathbb{I}_v = 0$.

A. First Order Algorithm

Set $\Phi_t(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|_2^2$, which is 1-strongly convex with respect to $\|\cdot\|_2$. And it is known that the dual norm of $\|\cdot\|_2$ is $\|\cdot\|_2$ itself, while $\Phi_t^* = \Phi_t$. Under these assumptions, we get the first order sparse online learning (FSOL) algorithm, which is the same with Regularized Dual Averaging (RDA) algorithm with soft 1-norm regularization [7].

Algorithm 2 First Order Sparse Online Learning (FSOL)

INPUT :λ, η
INITIALIZATION : $\theta_1 = 0$.
for $t = 1, \ldots, T$ **do**
  receive $\mathbf{x}_t \in \mathbb{R}^n$
  $\mathbf{w}_t = \text{sign}(\theta_t) \odot [|\theta_t| - \lambda_t]_+$
  predict $\hat{y}_t = \text{sign}(\mathbf{w}_t^\top \mathbf{x}_t)$
  receive $y_t \in \{-1, 1\}$ and suffer $\ell_t(\mathbf{w}_t) = [1 - y_t \mathbf{w}_t^\top \mathbf{x}_t]_+$
  $\theta_{t+1} = \theta_t + \eta L_t y_t \mathbf{x}_t$
end for

The regret of the previous first order algorithm is upper bounded $O(\sqrt{T})$:

$$R_T \leq \|\mathbf{w}\|_2 \sqrt{(X^2 + 2\lambda X)T} \tag{2}$$

B. Second Order Algorithm

Set $\Phi_t(\mathbf{w}) = \frac{1}{2}\mathbf{w}^\top A_t \mathbf{w}$, where $A_t = A_{t-1} + \frac{\mathbf{x}_t \mathbf{x}_t^\top}{r}, r > 0$ and $A_0 = I$. It is easy to verify that Φ_t is 1-strongly convex with respect to $\|\mathbf{w}\|_{\Phi_t}^2 = \mathbf{w}^\top A_t \mathbf{w}$. Its dual function $\Phi_t^*(\mathbf{w})$ is $\frac{1}{2}\mathbf{w}^\top A_t^{-1} \mathbf{w}$, while $\|\mathbf{w}\|_{\Phi_t^*}^2 = \mathbf{w}^\top A_t^{-1} \mathbf{w}$. Using the Woodbury identity, we can incrementally update the inverse of A_t as $A_t^{-1} = A_{t-1}^{-1} - \frac{A_{t-1}^{-1} \mathbf{x}_t \mathbf{x}_t^\top A_{t-1}^{-1}}{r + \mathbf{x}_t^\top A_{t-1}^{-1} \mathbf{x}_t}$. Under these assumptions, we get the second order sparse online learning (SSOL) algorithm. We can proof that the regret bound of the second order algorithm in an order of $O(\log(T))$.

Algorithm 3 Second Order Sparse Online Learning (SSOL)

INPUT :λ, η
INITIALIZATION : $\theta_1 = 0$.
for $t = 1, \ldots, T$ **do**
  receive $\mathbf{x}_t \in \mathbb{R}^n$
  $A_t^{-1} = A_{t-1}^{-1} - \frac{A_{t-1}^{-1} \mathbf{x}_t \mathbf{x}_t^\top A_{t-1}^{-1}}{r + \mathbf{x}_t^\top A_{t-1}^{-1} \mathbf{x}_t}$
  $\mathbf{u}_t = A_t^{-1} \theta_t$
  $\mathbf{w}_t = \text{sign}(\mathbf{u}_t) \odot [|\mathbf{u}_t| - \lambda_t]_+$
  predict $\hat{y}_t = \text{sign}(\mathbf{w}_t^\top \mathbf{x}_t)$
  receive $y_t \in \{-1, 1\}$ and suffer $\ell_t(\mathbf{w}_t) = [1 - y_t \mathbf{w}_t^\top \mathbf{x}_t]_+$
  $\theta_{t+1} = \theta_t + \eta L_t y_t \mathbf{x}_t$
end for

C. Diagonal Algorithm

Although the previous second order algorithm significantly reduced the regret bound than the first order algorithm, it will consume $O(d^2)$ time, which reduced its application to real-world high dimension problems. To keep the computational time still $O(d)$ similar with the traditional online learning, we further explored the diagonal algorithm, which will only maintains a diagonal matrix. Its details is in the Algorithm (4). In the following experiment, we mainly adopt the diagonal second order sparse online learning algorithm unless otherwise specified, which is also denoted as "SSOL".

Algorithm 4 Diagonal Second Order Sparse Online Learning

INPUT :λ, η
INITIALIZATION : $\theta_1 = 0$.
for $t = 1, \ldots, T$ **do**
  receive $\mathbf{x}_t \in \mathbb{R}^n$
  $A_t^{-1} = A_{t-1}^{-1} - \frac{A_{t-1}^{-1} \text{diag}(\mathbf{x}_t \mathbf{x}_t^\top) A_{t-1}^{-1}}{r + \mathbf{x}_t^\top A_{t-1}^{-1} \mathbf{x}_t}$
  $\mathbf{u}_t = A_t^{-1} \theta_t$
  $\mathbf{w}_t = \text{sign}(\mathbf{u}_t) \odot [|\mathbf{u}_t| - \lambda_t]_+$
  predict $\hat{y}_t = \text{sign}(\mathbf{w}_t^\top \mathbf{x}_t)$
  receive $y_t \in \{-1, 1\}$ and suffer $\ell_t(\mathbf{w}_t) = [1 - y_t \mathbf{w}_t^\top \mathbf{x}_t]_+$
  $\theta_{t+1} = \theta_t + \eta L_t y_t \mathbf{x}_t$
end for

V. Experiments

A. Experimental Setup

In our experiments, we compare the proposed algorithms with a set of state-of-the-art algorithms. The methodology

Table I
LIST OF COMPARED ALGORITHMS, "TG" MEANS TRUNCATE GRADIENT AND "DA" MEANS DUAL AVERAGING.

Algorithm	Order	Sparsity	Description
STG	1st	TG	Stochastic Gradient Descent [3]
FOBOS	1st	TG	FOrward Backward Splitting [5]
Ada-FOBOS	2nd	TG	Adaptive regularized FOBOS [19]
Ada-RDA	2nd	DA	Adaptive regularized RDA [19]
FSOL	1st	DA	The proposed Algorithm 2
SSOL	2nd	DA	The proposed Algorithm 4

details of these algorithms are listed in Table I.

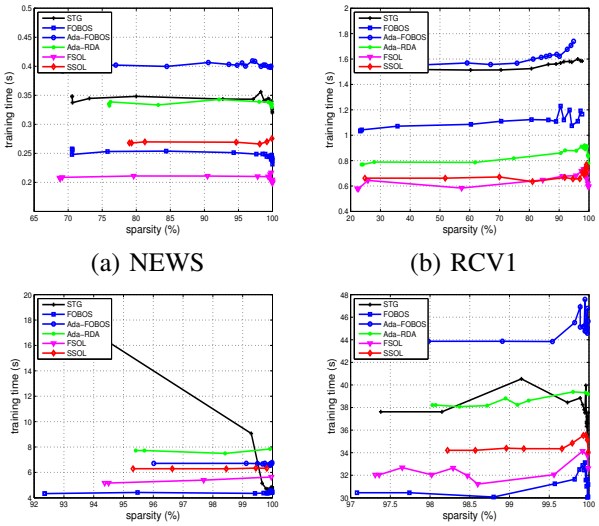
To examine the binary classification performance, we evaluate all the previous algorithms on a number of benchmark datasets from web machine learning repositories. Table II shows the details of all the datasets in our experiments. These datasets are selected to allow us evaluate the algorithms on various characteristics of data, in which the number of training examples ranges from thousands to millions, feature dimensionality ranges from hundreds to about 16-million, and the total number of non-zero features on some dataset is more than one billion. For the very large-scale WEBSPAM dataset, we run the algorithms only once. The sparsity as shown in the last column of the table denotes the ratio of non-active feature dimensions, as some feature dimensions are never active in the training process, which is often the case for some real-world high-dimensional dataset, such as WEBSPAM. For parameter tuning, we conduct a 5-fold cross validation to search the parameters with the fixed sparsity regularization parameter $\lambda = 0$ on each dataset.

B. Experiment on Error Rate

In this experiment, we compare the proposed algorithms (FSOL and SSOL) with the other algorithms on several real-world datasets. Table II shows the details of six datasets, which can be roughly grouped into two major categories: the first two datasets (AUT and PCMAC) are general binary small-scale datasets and the corresponding experimental results are shown in Figure 1 (a)-(b); and the rest four datasets (NEWS, RCV1, URL, and WEBSPAM) are large-scale high-dimensional sparse datasets and the corresponding experimental results are shown in Figure 1 (c)-(f). We can draw several observation from these results as follows.

First of all, we observe that most algorithms can learn an effective sparse classification model with only marginal or even no loss of accuracy. For example, in Figure 1 (d), the performances of all the algorithms are almost stable when sparsity level is smaller than 80%. It indicates that all the compared sparse online classification algorithm can effectively explore the low level sparsity information. Second, for most cases, we observe that there exists some sparsity threshold for each algorithm, below which test error rate does not change much; but when sparsity level is greater than the threshold, test error rate gets worse quickly. Third, we observe that the dual averaging based second order algorithms (Ada-RDA and SSOL) consistently outperform the other algorithms (STG, FOBOS, FSOL, and Ada-FOBOS), especially for high sparsity level. This indicates that the dual averaging technique and second order updating rules are effective to boost the classification performance. Finally, when the sparsity is high, an essential requirement for high-dimensional data stream classification tasks, the proposed SSOL algorithm consistently outperforms the other algorithms over all the evaluated datasets. For example, when the sparsity is about 99.8% for the WEBSPAM dataset (the total feature dimensionality is $16,609,143$), the test error rate of SSOL is about 0.3%, while the Ada-RDA is 0.4% and the Ada-FOBOS is 0.55%, as shown in Figure 1 (f).

C. Experiment on Running Time



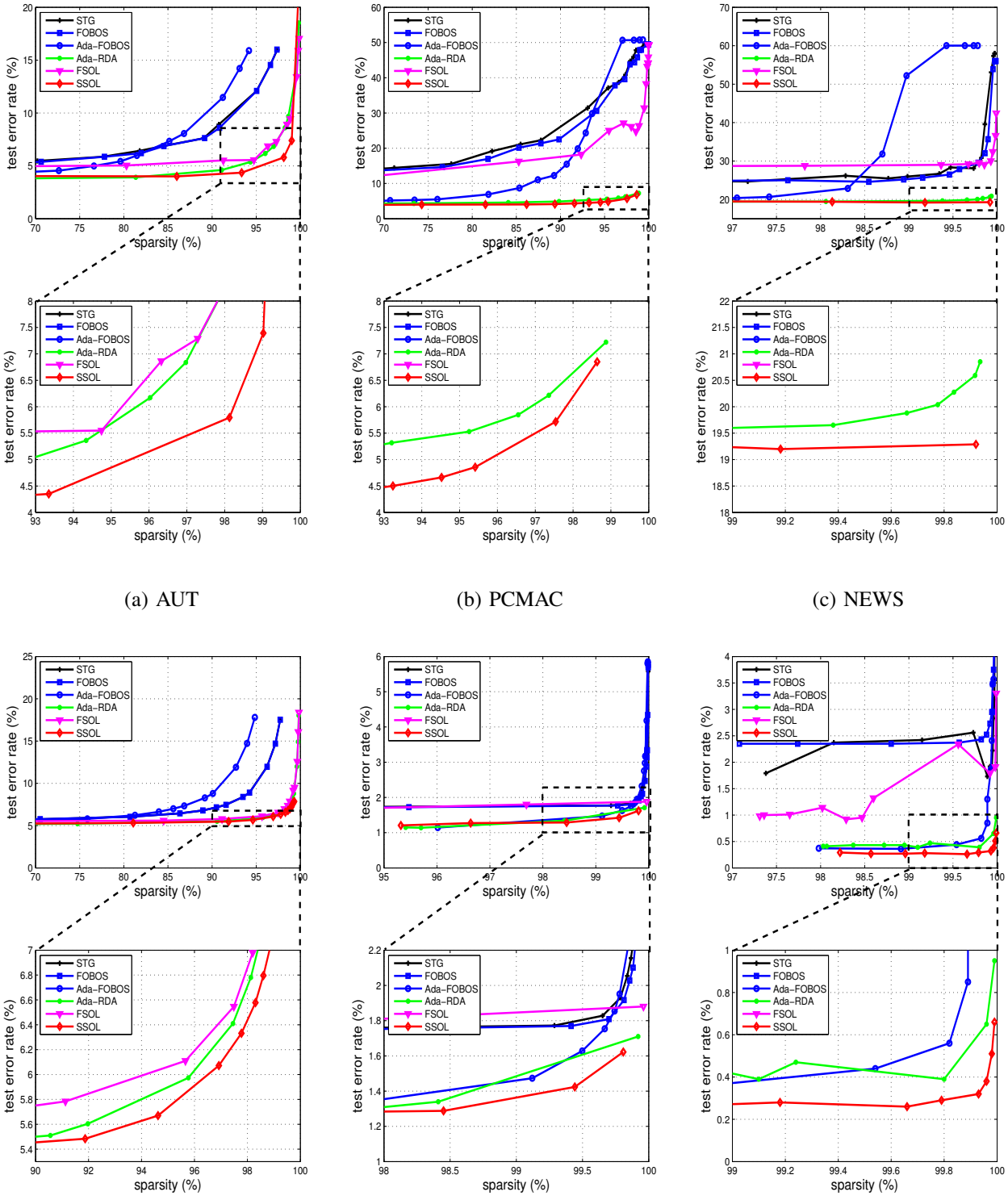
(a) NEWS (b) RCV1

(c) URL (d) WEBSPAM

Figure 2. Time cost on four large-scale datasets: NEWS, RCV1, URL, and WEBSPAM

We also examine time costs of different sparse online classification algorithms, and the experiment results are shown in Figure 2. In this experiment, we only adopt the four high-dimensional large-scale dataset. Several observations can be drawn from the results.

First of all, we observe that when the sparsity level is low, the time costs are generally stable; on the other hand, when the sparsity level is high, the time cost of the second algorithms sometimes will somewhat increase. For example, the test costs of Ada-FOBOS, Ada-RDA and FSOL in Figure 2 (b) & (d). One possible reason may be that when the sparsity level is high, the model might not be informative enough for prediction and thus may suffer significant more updates. Since second-order algorithms are



(a) AUT (b) PCMAC (c) NEWS

(d) RCV1 (e) URL (f) WEBSPAM

Figure 1. Test error rate on 6 large real datasets. (a)-(b) are two general datasets, (c)-(f) are four large-scale high-dimensional sparse datasets. The second and forth rows are the sub-figures of the first and the third rows with high sparsity level, respectively.

Table II
LIST OF REAL-WORLD DATASETS IN OUR EXPERIMENTS.

DataSet	#Train	#Test	#Feature Dimension	#Nonzero Features	Sparsity(%)
AUT	40,000	22,581	20,707	1,969,407	3.07
PCMAC	1,000	946	7,510	55,470	3.99
NEWS	10,000	9,996	1,355,191	5,513,533	29.88
RCV1	781,265	23,149	47,152	59,155,144	8.80
URL	2,000,000	396,130	3,231,961	231,259,917	7.44
WEBSPAM	300,000	50,000	16,609,143	1,118,443,083	95.82

more complicated than first-order algorithms, they are more sensitive to the increasing number of updates.

Second, we can see that the proposed SSOL algorithm runs more efficiently than another second-order based algorithms (Ada-RDA and Ada-FOBOS). It is even sometimes better than the first order based algorithm (e.g. FOBOS and STD). However, the first order FSOL algorithm is consistently faster than the second order SSOL algorithm.

In summary, the proposed SSOL algorithm can achieve comparable or even better performance than all the compared second-order algorithms with less time cost.

VI. CONCLUSIONS AND FUTURE WORK

In this paper we introduced a framework of sparse online classification (SOC) for large-scale high-dimensional data stream classification tasks. We first showed that the framework essentially includes an existing first-order sparse online classification algorithm as a special case, and can be further extended to derive new sparse online classification algorithms by exploiting second-order information. We analyzed the performance of the proposed algorithms on several real word datasets, in which the encouraging experimental results showed that the proposed algorithms are able to achieve the state-of-the-art performance in comparison to a large family of diverse online learning algorithms.

REFERENCES

[1] S. Radicati, "Email statistics report, 2013-2017." The Radicati Group,Inc., Tech. Rep., April 2013.

[2] S. Youn and D. McLeod, "Spam email classification using an adaptive ontology." *JSW*, vol. 2, no. 3, pp. 43–55, 2007.

[3] J. Langford, L. Li, and T. Zhang, "Sparse online learning via truncated gradient," *J. Mach. Learn. Res.*, vol. 10, pp. 777–801, 2009.

[4] S. C. Hoi, J. Wang, and P. Zhao, "Libol: A library for online learning algorithms," *Journal of Machine Learning Research*, vol. 15, pp. 495–499, 2014.

[5] J. Duchi and Y. Singer, "Efficient online and batch learning using forward backward splitting," *J. Mach. Learn. Res.*, vol. 10, pp. 2899–2934, 2009.

[6] X. Gao, S. C. Hoi, Y. Zhang, J. Wan, and J. Li, "Soml: Sparse online metric learning with application to image retrieval," *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2014.

[7] L. Xiao, "Dual averaging methods for regularized stochastic learning and online optimization," *The Journal of Machine Learning Research*, vol. 9999, pp. 2543–2596, 2010.

[8] S. Shalev-Shwartz and A. Tewari, "Stochastic methods for l 1-regularized loss minimization," *The Journal of Machine Learning Research*, pp. 1865–1892, 2011.

[9] J. Wang, P. Zhao, S. C. Hoi, and R. Jin, "Online feature selection and its applications," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1–14, 2013.

[10] Y. Nesterov, "Primal-dual subgradient methods for convex problems," *Mathematical programming*, vol. 120, no. 1, pp. 221–259, 2009.

[11] S. Lee and S. J. Wright, "Manifold identification in dual averaging for regularized stochastic online learning," *The Journal of Machine Learning Research*, vol. 98888, pp. 1705–1744, 2012.

[12] J. Lu, S. Hoi, and J. Wang, "Second order online collaborative filtering," in *Asian Conference on Machine Learning*, 2013, pp. 325–340.

[13] S. Balakrishnan and D. Madigan, "Algorithms for sparse linear classifiers in the massive data setting," *The Journal of Machine Learning Research*, vol. 9, pp. 313–337, 2008.

[14] M. Dredze, K. Crammer, and F. Pereira, "Confidence-weighted linear classification," in *ICML*, 2008, pp. 264–271.

[15] K. Crammer, M. Dredze, and F. Pereira, "Exact convex confidence-weighted learning," in *NIPS*, 2008, pp. 345–352.

[16] K. Crammer, A. Kulesza, and M. Dredze, "Adaptive regularization of weight vectors," *Machine Learning*, pp. 1–33, 2009.

[17] J. Ma, A. Kulesza, M. Dredze, K. Crammer, L. K. Saul, and F. Pereira, "Exploiting feature covariance in high-dimensional online learning," in *ICAIS*, 2010, pp. 493–500.

[18] J. Wang, P. Zhao, and S. C. Hoi, "Exact soft confidence-weighted learning," in *ICML*, 2012.

[19] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *The Journal of Machine Learning Research*, pp. 2121–2159, 2011.